小组成员以及贡献度：赵浩瀚 20% 张毅中20% 武晋荣20% 凌郑20% 陈劭谦20%

### 一、背景介绍

财务报告的分析，特别是年度报告，在量化金融领域中获得了广泛的关注。年度报告是详细的文件，提供了关于公司财务状况、管理策略和未来前景的重要信息。这些报告对于投资者、分析师和研究人员评估公司表现并做出明智决策至关重要。

在本项目中，我们将探讨沪深股市上市公司年度报告中包含的文本信息的预测能力。具体而言，我们将研究这些文本内容是否能够用来预测公司股票在未来一年的收益率。为此，我们将使用词袋模型（Bag of Words）、Word2Vec模型和Doc2Vec模型，对报告中的文本进行处理，并比较这些模型在预测股票收益率方面的效果。

本项目的目标是评估文本分析方法在财务预测中的应用潜力，并通过样本外均方误差的比较，确定哪种模型在预测股票收益率方面表现最佳。通过这些分析，我们希望揭示年度报告中的文本信息与公司未来股价表现之间的潜在联系。

##### 二、数据下载

###### 数据概述

根据作业要求，我们需要至少500份企业年报，考虑到中国A股市场的特点，市值较大企业的信息披露较为完善、财务造假风险较小，我们选择沪深300指数成分股作为股票域。

由于作业要求中涉及未来一年和未来两年的预测，因此我们需要预留2022年、2023年作为预测的年份，因此我们下载沪深300成分股2020和2021年的年报，其未来一年收益率分别对应2021年和2022年，未来两年收益率分别对应2022年和2023年。

###### 成分股数据

如上所述，我们从同花顺下载2024年8月4日沪深300指数的成分股数据，下载的数据如表1所示：

表 1 沪深300成分股数据

|  |  |  |
| --- | --- | --- |
| code\_full | name | code |
| 000001.SZ | 平安银行 | 000001 |
| 000002.SZ | 万科A | 000002 |
| 000063.SZ | 中兴通讯 | 000063 |
| 000100.SZ | TCL科技 | 000100 |
| 000166.SZ | 申万宏源 | 000166 |

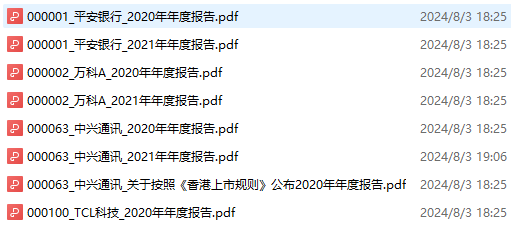
###### 年报数据

我们使用Python爬虫下载个股年报，使用巨潮资讯网，网址为：

<http://www.cninfo.com.cn/new/commonUrl/pageOfSearch?url=disclosure/list/search&lastPage=index>

通过爬虫，我们可以获取个股年报的下载链接，再使用Python的request命令下载对应链接下的文件即可。最终，下载好的文件如图1所示：

图 1 年报示例



###### 未来收益数据

如上所述，对于2020年的年报，我们下载2021年和2022年的收益率，对于2021年的年报，我们下载2022年和2023年的年报，最终结果如表2所示：

表 2 未来收益率

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| code\_full | name | code | 2021 | 2022 | 2023 |
| 000001.SZ | 平安银行 | 000001 | -0.1327 | -0.1763 | -0.2416 |
| 000002.SZ | 万科A | 000002 | -0.2213 | -0.0216 | -0.2670 |
| 000063.SZ | 中兴通讯 | 000063 | 0.0014 | -0.2102 | 0.0369 |
| 000100.SZ | TCL科技 | 000100 | -0.1009 | -0.3265 | 0.2130 |
| 000166.SZ | 申万宏源 | 000166 | -0.0103 | -0.1860 | 0.1069 |

###### **（五）数****据检验**

对于每一只股票，检验是否正确下载其年报文件。

如上所述，每一只股票应该对应2份年报，共600份，由于股票列表是2024年8月3日的数据，部分股票在2020、2021年可能还未上市，因此我们下载的文件数量为555份。

检查各文件名确认其为公司年报，发现如下几个异常情况：

1. 部分报告名为《×××持续督导年度报告》混入其中，故删除；
2. 部分公司的年报文件名以“年度报告全文”结尾，因此在之前爬虫的逻辑中并未考虑，由于年报数量已超过500份，故此部分年报未补充下载。
3. 部分公司年报包括境内（大陆）和境外（港股）的公示，我们认为这些报告均具有一定价值，因此保留了境外公示年报。

最终，我们一共有551份有效年报。

###### **（六）****数据处理**

对于下载好的公司年报数据，文件格式为pdf，我们对其进行分词，并转换为txt文件，为后续建模和预测做准备。

对于每一份pdf文件，我们按顺序进行如下处理：

1. 使用Python的fitz库读取pdf文件中的文本，过滤表格、图片等内容。
2. 使用Python的re正则表达式过滤文本中的空字符。
3. 使用Python的jieba对文本进行分词。
4. 根据停用词表去除停用词。
5. 保存为txt文件。

最终，得到的数据如图2所示：

图 2 文本数据



##### **三、词袋模型（Bag of Words, BoW）**

###### **（****一）词袋模型**

词袋模型是一种简单且常用的文本表示方法。它的核心思想是将文本中的词汇映射为固定长度的向量，而不考虑词汇的顺序或语法结构。每个向量的维度对应于词汇表中的一个词，值表示该词在文本中出现的频率。

在词袋模型中，每个文档都被表示为一个稀疏向量，其中每个元素的值为该文档中对应词汇的词频（TF）。尽管该模型没有考虑词与词之间的关系，但它在许多文本分类和信息检索任务中表现出色。词袋模型的简单性和可解释性使其在文本数据量较大但词汇关系复杂性较低的情况下尤为有效。

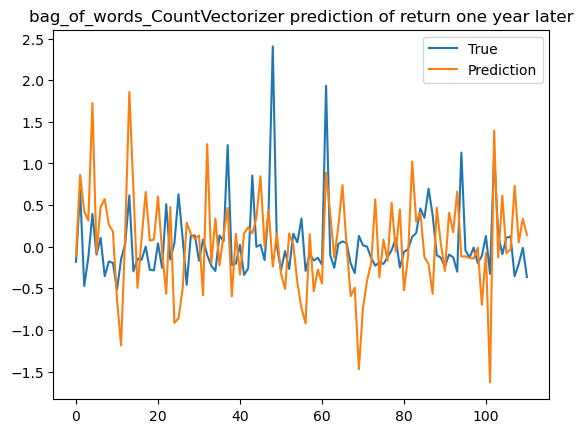
在代码实现中，首先加载了包含股票年收益率的数据集，并对每份年报文本进行读取和存储处理。具体操作是遍历每行数据，将对应的PDF格式的年报转换成文本文件，并保存到DataFrame中。这为后续文本分析提供了基础数据。

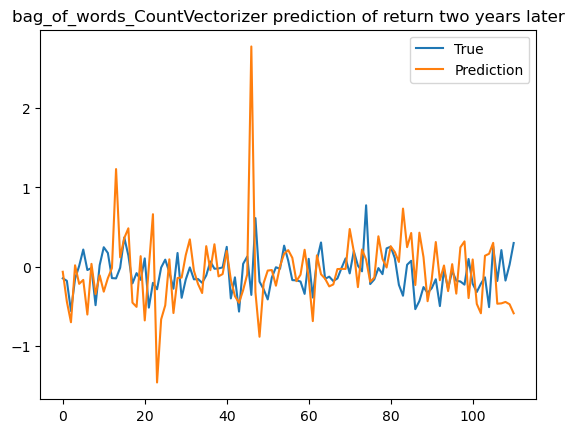
接下来，使用CountVectorizer对文本内容进行了向量化处理，将文本转化为词频的数值表示。这一步骤生成了一个稀疏矩阵，每一行对应一篇报告，每一列代表一个词汇的出现频率。

之后，该代码将得到的特征矩阵转换为数组格式，并将股票下一年与下两年收益率作为目标变量y。为了评估模型的效果，数据被分割为训练集和测试集，具体分割比例为80%的数据用于训练，20%的数据用于测试，确保了模型可以在未见数据上进行有效的性能评估。

模型训练使用线性回归算法，通过训练集数据拟合后，对测试集进行预测，并计算了预测结果的均方误差（MSE），以量化模型在预测未来收益率方面的准确性。最终，代码通过绘图将实际收益率与预测收益率进行了可视化对比，直观地展示了模型的预测能力。

从结果上来看，使用词袋模型对股票下一年收益率的预测mse为0.4266633929825962，对第二年股票收益率预测的mse为，下面给出真实值与预测值的对比图，结果上第二年的预测效果更好些。

****

****

###### **（二）TF-IDF 模型**

由于词袋模型与TF-IDF模型具有较高相似性，且TF-IDF模型是基于词袋模型的改进模型。TF-IDF模型通过引入IDF，以更好地捕捉文本的特征。词袋模型仅仅考虑文本中每个词出现的频率，而不考虑词的顺序或位置。TF-IDF不仅考虑词在文本中出现的频率（TF），还考虑了该词在整个语料库中的逆文档频率（IDF）。考虑到这种相似性，小组也针对TF-IDF模型进行了模型训练并预测了未来一年与未来两年的股票收益率情况。

TF-IDF（Term Frequency-Inverse Document Frequency）模型是一种增强版的词袋模型，它不仅考虑了词汇在文档中的频率（Term Frequency, TF），还结合了词汇在整个语料库中的逆文档频率（Inverse Document Frequency, IDF）。通过这种方式，TF-IDF能够降低常见词汇的权重，并增强对文档区分度更高的词汇的影响。

在TF-IDF模型中，每个词的TF-IDF值由其在特定文档中的频率（TF）与其在整个文档集中的反频率（IDF）相乘得出。较高的TF-IDF值通常表明该词在特定文档中具有较强的代表性，且在其他文档中较为罕见。

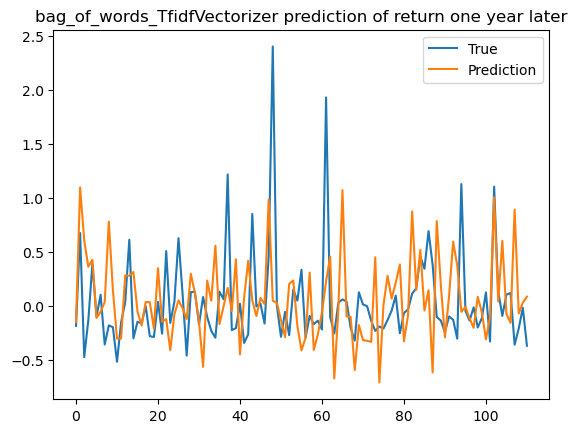
在代码实现中，首先加载了包含股票年收益率的数据集，并对每份年报文本进行读取和存储处理。具体操作是遍历每行数据，将对应的PDF格式的年报转换成文本文件，并保存到DataFrame中。这为后续文本分析提供了基础数据。

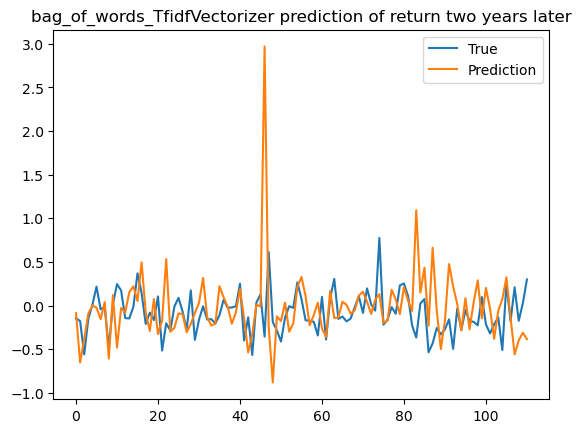
接下来，使用TfidfVectorizer对文本内容进行了向量化处理，将文本转化为TF-IDF的数值表示。这一步骤生成了一个稀疏矩阵，每一行对应一篇报告，每一列代表一个词汇的加权频率，体现了词汇在文档中的重要性及其在整个语料库中的分布。

之后，该代码将得到的特征矩阵转换为数组格式，并将股票下一年与下两年收益率作为目标变量y。为了评估模型的效果，数据被分割为训练集和测试集，具体分割比例为80%的数据用于训练，20%的数据用于测试，确保了模型可以在未见数据上进行有效的性能评估。

模型训练使用的是线性回归算法，通过训练集数据拟合后，对测试集进行预测，并计算了预测结果的均方误差（MSE），以量化模型在预测未来收益率方面的准确性。最终，代码通过绘图将实际收益率与预测收益率进行了可视化对比，直观地展示了模型的预测能力。

从结果上来看，使用词袋模型对股票下一年收益率的预测mse为，对第二年股票收益率预测的mse为，下面给出真实值与预测值的对比图，结果上第二年的预测效果更好些。

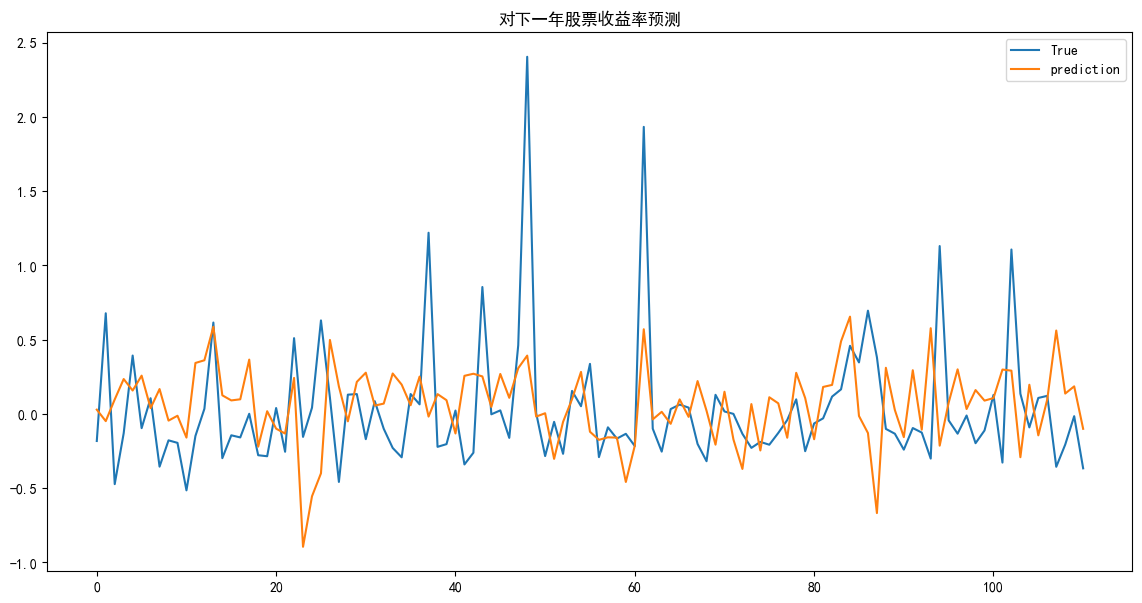


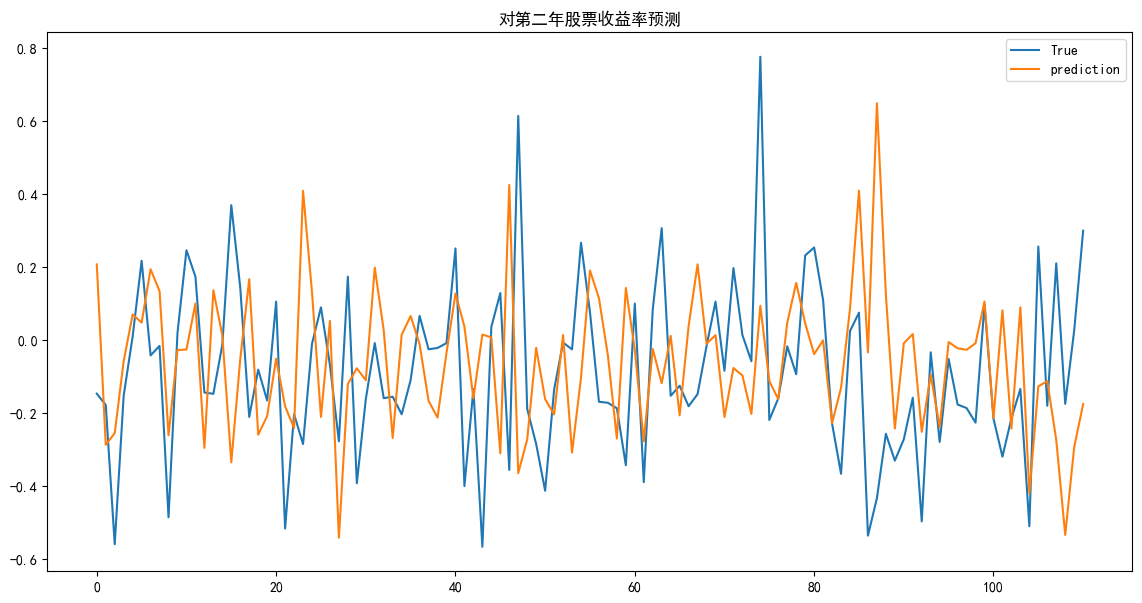


##### **四、Word2Vec模型**

Word2Vec是一种将词汇映射到低维向量空间的神经网络模型，最初由Google在2013年提出。其核心思想是将每个词表示为一个固定大小的向量（即词嵌入），这些向量能够捕捉词汇之间的语义关系。在向量空间中，语义相似的词通常会有相似的向量表示。Word2Vec有两种主要的训练方法：CBOW（Continuous Bag of Words）和Skip-Gram。CBOW模型：它根据上下文词预测目标词。例如，给定一个句子，CBOW模型会尝试预测其中的每个词的上下文。这种方法在文本数据量较小时通常表现较好，因为它更关注全局的语义信息。Skip-Gram模型：与CBOW相反，Skip-Gram模型根据目标词预测其上下文词。也就是说，给定一个词，模型会尝试预测在其前后可能出现的词。Skip-Gram通常在大规模数据集上表现更佳，因为它更关注细粒度的词对关系。Word2Vec模型的目标是通过优化使相似词的向量在向量空间中距离更近。例如，“king”和“queen”的词向量可能在某个维度上非常接近，而与“apple”的距离较远。这种语义信息使得Word2Vec能够被广泛应用于文本分类、信息检索、问答系统等任务。训练Word2Vec模型的过程实际上是通过调整模型的权重，使得词与上下文之间的关系符合训练数据中的实际情况。通过反向传播和梯度下降，模型逐步调整词向量，使它们能够捕捉到上下文中的语义关系。训练完成后，模型可以通过多种方式使用，例如获取词的向量表示、计算词之间的相似度、查找与给定词最相似的词等。

在代码具体实现过程中，首先加载了股票收益率数据，并读取每份报告的文本内容。通过将报告文件从PDF转换为文本文件并存储在DataFrame中，程序为后续的文本处理打下了基础。接下来，使用gensim库中的Word2Vec类对报告文本进行训练。训练时，指定词向量维度为100，窗口大小为5，最小词频为1，以确保所有词语都参与模型训练。训练完Word2Vec模型后，代码计算了每个报告的平均词向量。具体来说，通过将每个词的词向量取平均值，生成一个固定维度的向量来表示整篇报告。这些向量随后被用作线性回归模型的特征输入，目标变量则是股票的下一年或第二年的收益率。在此基础上，将数据按照4：1的比例划分为训练集和测试集，使用训练集对线性回归模型进行拟合，并在测试集上进行预测。最后，计算并输出均方误差（MSE），以评估模型在样本外的预测性能。此外，代码还通过绘制真实值和预测值的对比图，直观展示了模型的表现。

使用Word2Vec模型对股票下一年收益率预测的MSE约为0.219，，而对第二年股票收益率预测的MSE约为0.089，下面给出了真实值和预测值的对比图，可以看到对第二年股票收益率进行预测的效果要明显更好一些。



##### **五、Doc2Vec模型**

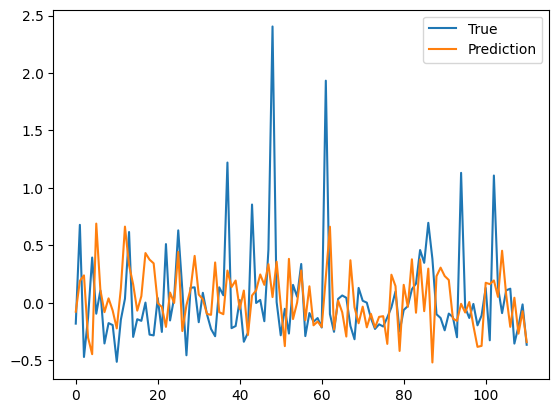
Doc2Vec方法是一种无监督算法，能从变长的文本（例如：句子、段落或文档）中学习得到固定长度的特征表示。Doc2Vec也可以叫做 Paragraph Vector、Sentence Embeddings，它可以获得句子、段落和文档的向量表达，是Word2Vec的拓展，可以不用固定句子长度，接受不同长度的句子做训练样本。

Doc2Vec模型是受到了Word2Vec模型的启发。Word2Vec预测词向量时，预测出来的词是含有词义的，Doc2Vec中也是构建了相同的结构，所以Doc2Vec克服了词袋模型中没有语义的缺点。假设现在存在训练样本，每个句子是训练样本，和Word2Vec一样，Doc2Vec也有两种训练方式，一种是分布记忆的段落向量（Distributed Memory Model of Paragraph Vectors , PV-DM）类似于Word2Vec中的CBOW模型，另一种是分布词袋版本的段落向量（Distributed Bag of Words version of Paragraph Vector，PV-DBOW）类似于Word2Vec中的Skip-gram模型。

Doc2vec相对于Word2vec不同之处在于，在输入层增添了一个新的句子向量Paragraph vector，Paragraph vector可以被看作是另一个词向量，它扮演了一个记忆角色。Average Word Vectors中，使用Word2Vec训练词向量，因为每次训练只会截取句子中一小部分词训练，而忽略了除了本次训练词以外该句子中的其他词，这样仅仅训练出来每个词的向量表达，句子只是每个词的向量累加在一起取平均的一种表达。正如上面所说的Average Word Vectors的缺点，忽略了文本的词序问题。而Doc2vec中的Paragraph vector则弥补了这方面的不足，它每次训练也是滑动截取句子中一小部分词来训练，Paragraph Vector在同一个句子的若干次训练中是共享的，所以同一句话会有多次训练，每次训练中输入都包含Paragraph vector。它可以被看作是句子的主旨，有了它，该句子的主旨每次都会被作为输入的一部分来训练。这样每次训练过程中，不光是训练了词，得到了词向量。同时随着一句话每次滑动取若干词训练的过程中，作为每次训练的输入层一部分的共享Paragraph vector，该向量表达的主旨会越来越准确。

在代码具体实现过程中，首先加载了股票收益率数据，并读取每份报告的文本内容。通过将报告文件从PDF转换为文本文件并存储在DataFrame中，并将文本数据转换为gensim的TaggedDocument格式，这是训练Doc2Vec模型所需的格式。接着训练Doc2Vec模型，设置向量大小为100，窗口大小为5，最小词频为1，并使用4个工作线程。在模型训练之后，定义一个函数compute\_document\_vector，用于计算文档的向量表示，计算文档向量并将其存储在DataFrame的doc\_vector列中，再准备特征矩阵X和目标变量y，移除NaN值以确保数据的一致性。在此基础上，将数据按照4：1的比例划分为训练集和测试集，使用训练集对线性回归模型进行拟合，并在测试集上进行预测。最后，计算并输出均方误差（MSE），以评估模型在样本外的预测性能。此外，代码还通过绘制真实值和预测值的对比图，直观展示了模型的表现。

具体真实值和预测值的对比图如下：



##### **六、模型预测性能比较**

1. **不同模型性能比较**

在上述提及的四大模型之间我们比较了各个模型对第一年收益率以及第二年收益率的预测情况，通过对不同模型在验证集的预测进行比较（见下图）可以发现如下结论：

1 Bag of Words (CountVectorizer) 和 TF-IDF Vectorizer 在两个时间窗口中的表现都较差。第一年预测MSE分别为0.427与0.275，第二年预测MSE分别为0.249与0.216。

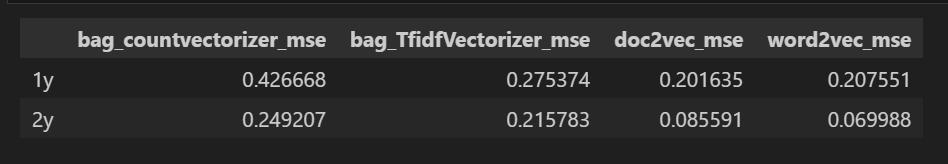
原因: 这两种模型都是基于词频的简单文本表示方法。它们无法捕捉词汇之间的语义关系，也不能理解文本的上下文。特别是在处理涉及复杂金融文本的任务时，这种方法的局限性显得尤为明显。TF-IDF 在一定程度上考虑了词频的重要性，与传统词袋模型相比模型表现有所提高，但仍然没有克服 Bag of Words 的基本局限性。

2 Doc2Vec 和 Word2Vec 模型表现较好，两个模型两年的MSE都接近，且都小于Bag of Words (CountVectorizer) 和 TF-IDF Vectorizer 模型。

原因: 这两个模型能够捕捉到词汇之间的语义关系。Doc2Vec 通过将整个文档表示为一个向量，能够更好地理解文本的整体语义，这对于金融文档的长期预测尤其重要。Word2Vec 则通过学习词汇的分布式表示，能够更好地捕捉词汇的上下文信息。相较于词袋模型，这些模型更能应对金融文本的复杂性和非线性特征。

3 Doc2Vec 和 Word2Vec 在预测两年后收益率时的 MSE 显著降低。

原因: 这可能是因为这些模型能够更好地捕捉文本中的长期趋势和语义关系，因此在预测较长时间范围内的股票收益时表现得更为准确。此外，较长时间范围的预测可能削弱了市场短期波动的影响，使得模型可以更专注于文本中所反映的长期趋势和信息。



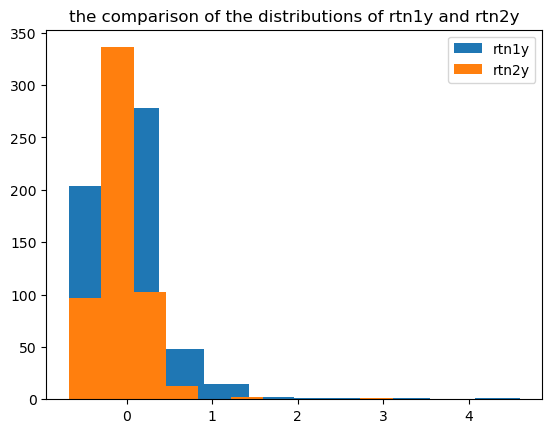
图：四大模型样本外MSE比较

1. **预测性能****差异原因**
2. **收益率分****布的分析**

不同年份收益率的分布会对模型的预测性能产生影响，通过观察下面rtn1y与rtn2y的分布可以发现二者的分布有明显的差异：

· **第一年收益率 (rtn1y)**：收益率集中在接近0的区域，右尾较长。这表明大多数公司的年收益率接近于0，但存在少数公司收益率较高。

· **第二年收益率 (rtn2y)**：收益率也集中在接近0的区域，但分布更加集中，尾部较短。这表明大多数公司的两年收益率变化较小，市场波动趋于稳定。



1. **模****型对不同年份收益率的预测性能**

**第一年预测性能差异：**由于第一年收益率分布具有较长的右尾，市场波动较大，这种波动性增加了预测的难度。传统的文本向量化方法（如 Bag of Words 和 TF-IDF）在捕捉这种复杂的非线性关系时表现较差，而语义模型（如 Word2Vec 和 Doc2Vec）由于其更强的语义理解能力，表现相对更好。

**第二年预测性能提高：**两年后的收益率分布更加集中且尾部较短，表明市场的波动性减少。这可能是由于市场的长期趋势变得更加稳定，导致模型在第二年预测时面临更简单的模式。这种情况有利于所有模型的预测性能提升，尤其是那些能够捕捉长期趋势的模型（如 Doc2Vec 和 Word2Vec）。

1. **收益率分布对模型预测性能的影响**

**Bag of Words 和 TF-IDF：**这些模型仅能捕捉词频信息，对复杂的市场波动（如第一年）无法有效建模，导致较高的MSE。对于更稳定的第二年收益率，这些模型表现有所提升，但仍然受到其固有限制的影响。

**Doc2Vec 和 Word2Vec：**由于它们能够捕捉到文本中的语义信息和长期趋势，尤其适用于处理复杂和非线性的数据。因此，在市场波动较大的第一年预测中表现相对更好，而在第二年更加稳定的环境下，表现进一步提升。

1. **数据本身的影响**

**市场特征和数据周期：**从收益率的分布来看，第一年可能存在更大的市场波动或不确定性因素，而第二年市场则趋于平稳。这种波动性和不确定性影响了模型的预测难度，也直接导致了模型预测性能的差异。